ICANN85 Mumbai | CF – How it Works: DNS Fundamentals
Sunday, March 8, 2026 – 16:30 to 17:30 IST

FERNANDA IUNES

All righty. Thank you very much. Hello, and welcome to How It Works: DNS Fundamentals. My name is Fernanda Iunes, and I'm a participation manager for the session. Please note that the session is being recorded and is governed by the ICANN Community Participant Code of Conduct, ICANN Expected Standards of Behavior, and the ICANN Community Anti-Harassment Policy.

Please observe the following guidelines to participate in the session. I've also posted them in the chat for your reference. During the session, questions will only be read aloud if submitted within the Q&A pod. Interpretation for this session will include English, French, and Spanish. If you would like to speak during this session, please raise your hand in Zoom. When called upon, virtual participants will be given permission to unmute in Zoom. On-site participants will use a physical microphone to speak. Only questions posted in the Q&A pod will be read aloud during the session as time permits and when directed by the chair of the session. Please state your name for the record, the language you will speak if speaking a language other than English, and speak clearly and at a moderate pace. With that, I'll hand it back over to Siranush. Thanks.

SIRANUSH VARDANYAN: Thank you, Fernanda. We are coming to the end, to the last session, and we will be talking about DNS fundamentals. You already know my colleague, Champika, and my colleague, Nico, who is joining us from Uruguay remotely. Without further ado, let's give the floor to Champika to talk about DNS fundamentals. If you have any questions, feel free to ask those either here using your mic or remotely in the chat. Champika, the floor is yours.

CHAMPIKA WIJAYATUNGA: Thank you, Siranush. Namaste, and good evening to all of you. To introduce myself again, I'm Champika. I'm part of the technical engagement team covering the APAC region. This morning, we spoke about internet identifiers. As part of that, we touched on IP addresses, autonomous system numbers, and domain names. In this session, we will go into more detail regarding domain names and discuss some security aspects. That is the plan.

Sorry for the little delay; I was logged out of the system, so I had to re-login. I think I'm back. Let me share the slides.

FERNANDA IUNES: The system is going to log you out right when you need to share.

CHAMPIKA WIJAYATUNGA: Yeah.

SIRANUSH VARDANYAN This is the way.

CHAMPIKA WIJAYATUNGA Finally. Thank you for your patience. I was saying that this session is going to be on DNS fundamentals, so we will discuss DNS and its security aspects. I think we have been introduced, so I will skip that. Let us discuss the namespace. DNS is the Domain Name System. The first early drafts of DNS RFCs came during the early 1980s. It has been existing for over forty years.

DNS is a distributed database. The purpose of the Domain Name System is to map IP addresses to a resource that we can easily remember. Even this morning, we spoke about IP addresses. As far as data transfer is concerned, when you want to send data from one host to another, we could still send it with the IP addresses. However, as humans, it is difficult for us to remember them. If you remember those IPv6 addresses we discussed, they are not easy to remember. That is why we have a mechanism to map these IP addresses into a resource that we can easily recall.

The Domain Name System is a well-structured distributed system. There are concepts like delegations that we have to consider. When considering this distributed database, we speak about the namespace. When you think of the namespace, you can see a hierarchical view of the Domain Name System. At the top of the hierarchy, we have the root. Below that, we have top-level

domains, then second-level domains, and so on, all the way down. This is a space, or simply a database.

When I say namespace, I mean the entire database we are talking about. For example, if you are talking about the root namespace, there is a certain space the root is responsible for. The root can delegate part of that responsibility to another party. This delegation means giving responsibility to another party who must then look after that specific namespace. The root delegates top-level domains, and the top-level domains delegate second-level domains. If you follow the hierarchy, you can see how this happens. If I am the root, I would delegate .com to you. If I get a query for www.example.com, I do not have to answer the full question. I would give a referral to .com because it has been delegated.

If there are any engineers here doing DNS configurations, you may have to know about Fully Qualified Domain Names. These labels end with a dot. That trailing dot refers to the root because it is the absolute path. There are always parent-child relations where the root is the parent and the TLD is the child. There is also the terminology of zones. A domain includes everything below a certain node in the hierarchy. A zone is an administrative space. If I delegate example.com to someone else, I keep an administrative space so that I can provide references to their name servers. The zone is simply the administrative space, whereas the domain is everything below yourself.

Inside the database or zone, we have resource records, or RRs. These are the mappings between resources. In DNS, we map a name to a number. Common resource records include A records for IPv4 addresses, AAAA records for IPv6 addresses, and NS records to define name servers. The SOA, or Start of Authority, defines administrative information for the zone. This includes name server details, email contacts, and timing parameters for secondary servers. We need secondary servers for redundancy. These servers must be synchronized to remain consistent. All this synchronization happens based on the timing parameters defined in the SOA record. MX records define your mail servers. If your DNS servers are down, you will have a problem receiving emails because the mail servers are defined in your DNS database.

Earlier, I mentioned delegations. If I am the root, I can delegate .com to you. I need to define your name server records in my database. Any query that comes for .com will be referred to the .com name servers, such as a.gtld-servers.net. In your organization, you may only have two name servers, so I would have two NS records for you. Inside the .com zone, they would do the same thing to delegate example.com to the next level. We also have glue records. If there are name servers in the same hierarchy, we must define their IP addresses in the zone. If they are not in the same hierarchy, we can get those IP addresses through usual DNS resolution.

Regarding root zone management, the root servers keep the root database. Changes to the root zone are coordinated by IANA. IANA

has a contract with Verisign to perform the updates. If a TLD manager wants to change their name servers, they send a request to IANA. IANA coordinates with Verisign to update the root zone file. This involves updating the database, creating the new file, publishing it, and transferring it to the root server operators. There are 12 organizations operating the 13 root server identities. They operate the servers so that DNS resolvers can get referrals. Root server operators do not edit the root zone file themselves.

There are nearly 2,000 root server instances scattered around the world, and they all carry the same root zone file. If you change your name server IP address at the ccTLD level, that change must be provided to IANA to be distributed to all root server instances. Regarding DNS as a service, a stub resolver is a piece of code in your operating system that communicates with applications like browsers. The stub resolver contacts a recursive resolver. Typically, internet service providers, network operators, and mobile operators operate these resolvers, which are also called caching servers. The recursive resolver then talks to authoritative name servers. Authoritative servers are those authorized to manage a specific zone.

If I want to visit www.example.com, my browser sends a request to the stub resolver. The stub resolver sends it to the recursive resolver. If the recursive server does not have the information in its cache, it goes to the root name servers. The resolver knows how to find the root name servers through a configuration file. Using Anycast technology, the request is sent to the most optimal root

server. The root server responds with the name servers for the top-level domain. The resolver then contacts the .com name servers, which refer it to the example.com name servers. Finally, the example.com server provides the IP address for the web server. The recursive resolver can remember this answer for a certain amount of time, which we call caching.

The Time to Live, or TTL, is defined by the authoritative server administrator. If you define a TTL for one day, the resolver will answer from its memory for that period. You must balance TTL timing; if you change your zone daily, you do not want very long TTL values. Once the client has the IP address, it can talk directly to the web server to download the page. We must consider if this process is secure. Someone could pollute the memory through cache poisoning, or someone could pretend to be your resolver and provide a malicious IP address for a banking site. This is why we need security.

DNS queries usually go in plain text, making them easy to intercept. If someone needs privacy, they can use encryption protocols like DOT or DOH. We also need to ensure authenticity and integrity. Authenticity verifies that you are talking to the right server, while integrity ensures the data was not modified. We achieve this through DNSSEC. DNSSEC uses digital signatures to protect DNS data. We use asymmetric keys, keeping private keys safe and publishing public keys in the zone. While authoritative name servers perform the signing, recursive resolvers perform the

validation. Trust is propagated from the child to the parent all the way to the root.

All these red crosses represent weak points in the DNS infrastructure. To protect them, we use encryption, DNSSEC, and secure transactions. We also have a framework called KINDNS. You can visit KINDNS.org to see good practices for securing DNS infrastructure. I will stop here and take questions. Thank you.

SIRANUSH VARDANYAN: We have so many questions, but I'm sure we will not be able to take them all as we only have four minutes left. Champika, you will need to stay a couple of minutes more.

CHAMPIKA WIJAYATUNGA: Yes, sure. I am happy to stay and answer any questions.

SIRANUSH VARDANYAN: Jasbindar Singh.

JASBINDAR SINGH: Thank you, Champika. My name is Jasbindar Singh from Fiji, and I am an ICANN 85 fellow. My question is on the root zone maintainer. You said Verisign is the current maintainer. Can you briefly elaborate on the criteria for selecting a maintainer, and what continuity measures are in place if the current maintainer fails?

CHAMPIKA WIJAYATUNGA Good question, but we have John here. John is the one who can answer that.

JOHN CRAIN I am John Crain, ICANN Chief Technology Officer. We have been working with the same contractor since before DNSSEC, going back many years. There is a contract with security clauses and requirements for redundancy. The contract regularly comes up for review, but I do not think it has been put out for bid in a long time. This is not like getting someone to maintain your car; this must remain as stable as possible. It was also part of the agreement when we transitioned ICANN away from the U.S. government that we would use a specific maintainer. The contract is public and available.

SIRANUSH VARDANYAN Thank you, John. Altangerel, please.

ALTANGEREL ALTANKHUYAG Thank you. My name is Altangerel from Mongolia, an ICANN 85 fellow. The DNS root server system has 13 identities labeled A through M and is operated by 12 organizations. For a specific region, how is the decision made about which root server instances are deployed there?

CHAMPIKA WIJAYATUNGA: From an ICANN perspective, we are the root server operator for L root. Other operators include the Internet Systems Consortium for F and NetNode for I. You can meet these operators at RSSAC meetings. If you want an L root instance in your network, you can talk to ICANN engagement staff to go through the criteria.

We have deployed many L root instances globally. Policies can differ between operators. ICANN usually expects the hosting party to provide the hardware, but other operators might sponsor the hardware. You can talk to any of the operators to see how to have an instance in your country.

ALTANGEREL ALTANKHUYAG: Thank you, Champika.

CHAMPIKA WIJAYATUNGA: Thank you.

SIRANUSH VARDANYAN: Thank you, Champika, and thank you very much for being here for this last session. This was a very interesting session. If you still have questions for Champika, please stay for another 10 to 15 minutes. We are finished for today. Tomorrow we start with the welcome ceremony for the official ICANN 85 start day. We will see the welcome ceremony and the gala event organized by the host in this venue. I look forward to seeing you all there. This meeting is adjourned. Thank you.

CHAMPIKA WIJAYATUNGA Thank you very much.

[END OF TRANSCRIPTION]